(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 716 371 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
12.06.1996 Bulletin 1996/24

(51) Int. Cl.[6]: **G06F 3/12**

(21) Application number: **95119135.2**

(22) Date of filing: **05.12.1995**

(84) Designated Contracting States:
DE FR GB IT NL

(30) Priority: **06.12.1994 JP 301723/94**

(71) Applicant: **CANON KABUSHIKI KAISHA**
Tokyo (JP)

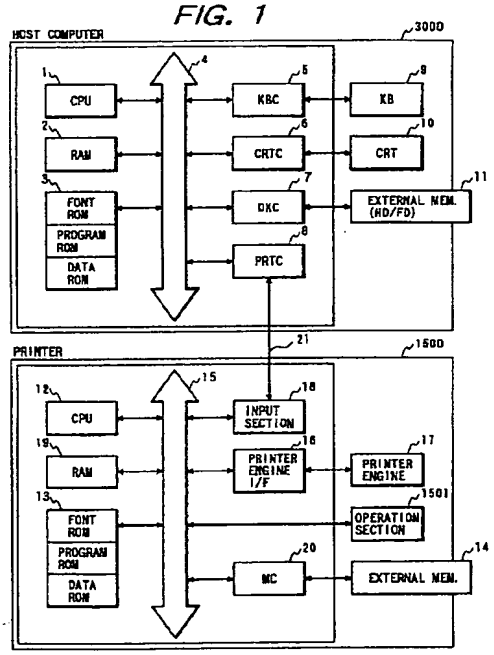
(72) Inventor: **Nakagiri, Koji,**
c/o Canon K.K.
Ohta-ku, Tokyo (JP)

(74) Representative: **Tiedtke, Harro, Dipl.-Ing.**
Patentanwaltsbüro
Tiedtke-Bühling-Kinne & Partner
Bavariaring 4
80336 München (DE)

(54) Information processing apparatus having automatic OS selecting function

(57) There is provided an information processing apparatus which is constructed by a host computer and a peripheral device connected to the host computer through a bidirectional interface and which inputs and outputs data between the host computer and the peripheral device through a device driver that corresponds to the peripheral device and is loaded in the OS of the host computer. An output device such as a printer which is controlled by the host computer has an identifying unit to identify the kind of OS of the host computer and a controller to output a device driver for controlling the device to the host computer on the basis of the identification result. The peripheral device has a memory in which the device driver corresponding to the peripheral device has been stored, the device driver in the memory is transferred to the host computer as necessary and is loaded into the OS of the host computer.

FIG. 1



EP 0 716 371 A2

Description

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to an information processing apparatus and, particularly, to an information processing apparatus for inputting and outputting data from/to a device connected through a device driver (program) loaded in an OS of a host computer.

Related Background Art

Hitherto, various products are used as OSs (operating systems) of a host computer. There is a case where among OS software, a portion (device driver) for driving devices such as printer, display, and the like is compiled integrally with a kernel of the OS. In recent years, however, the device driver is often constructed as what is called a loadable module that is read in accordance with a set file when the OS is booted or the like. There is also a device driver having a format of an application program so as to enable another device driver to be activated during the operation of the OS.

Recently, the device driver (program) particularly for a printer is called a printer driver, a printer manager, or the like (hereinafter, simply referred to as "printer driver"). In many conventional printer drivers, data is inputted and outputted from/to a printer port at a very low level. However, there is also considered a printer driver for translating between a page description language (printer control language) that is understood by the printer and another printer control language or a format in accordance with the OS.

There is also a printer driver which have therein softwares corresponding to a plurality of devices (not limited to the printer, but including a plotter and the like as well) and which enables various output devices such as printer and the like to output by designating a kind of the printer, or the like.

In the OS using the loadable device driver, a device which has not initially been supported by the OS can be used by replacing the printer driver. Recently, when the output device such as a printer or the like is purchased, there is a case where an exclusive-use device driver for the device in the major OS is annexed.

A number of OSs for controlling a host computer also exist nowadays. The same host computer is controlled by different OSs or a plurality of OSs are installed in the same host computer and one of the OSs is activated by switching them. When the OSs are different, since the description formats of the device drivers are also different, it is necessary to supply the device driver every OS even in case of the printer drivers for driving the same printer as a target.

On the other hand, recently, on the output device side as well, a method whereby the printers are connected to a plurality of printer ports or a method whereby printers are switched and connected by using a change-over switch is used in order to selectively use the devices in accordance with obj cts.

When the printer to be used is changed as mentioned above, it is naturally necessary to load the printer driver corresponding to the printer and to change the setting of the printer driver, and the like. There is a case such that the printer drivers are different every OS as mentioned above or exist as different files on a disk, so that there are problems such that works for correctly loading or changing the printer driver by the user are very complicated and troublesome.

When the designation of the user is wrong, the data cannot be correctly outputted as a matter of course.

In the case where computers are connected to a network in which a number of printer devices exist, it is necessary to store all of the printer drivers in all of the computers on the network, so that there is a problem such that resources of external memories (hard disk and the like) are vainly consumed.

Although the above problems have been described with respect to the printer as an example, similar problems also occur even in device drivers of other peripheral devices such as display, plotter, sound card, PCMCIA card, and the like.

SUMMARY OF THE INVENTION

The invention is made to solve the above problems and it is an object of the invention to provide an information processing apparatus which can automatically load a device driver (program) adaptive to a peripheral device that is used into an OS without an operation of the user and which can prevent the use of a wrong device driver and that resources such as memory, disk, and the like of a computer are vainly used.

In order to accomplish the above object, according to the invention, there is provided an information processing apparatus which is constructed by a host computer and a peripheral device connected to the host computer through a bidirectional interface (for instance, bicentronics) and which inputs and outputs data between the host computer and the peripheral device through the device driver that corresponds to the peripheral device and is loaded in the OS of the host computer, wherein the peripheral device has memory means in which the device driver corresponding to the peripheral device has been stored, and the device driver in the memory means is obtained from the peripheral device as necessary and is loaded into the OS of the host computer.

By such a construction, the device driver prepared for the peripheral device can be obtained from the peripheral device as necessary and can be automatically loaded into the OS.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram for explaining a construction of a computer system to which the invention is applied;

Fig. 2 is a block diagram for explaining a transfer of a printer driver in a printer in Fig. 1;

Fig. 3 is a flowchart showing a processing procedure for reception data in the printer in Fig. 1;

Fig. 4 is a flowchart showing a transfer processing procedure for the printer driver in the printer in Fig. 1;

Fig. 5 is a block diagram for explaining a storage or loading of the printer driver in a host computer in Fig. 1;

Fig. 6 is a flowchart showing a using method of the printer driver in an OS of the host computer in Fig. 1; and

Fig. 7 is a flowchart showing a different using method of the printer driver in the OS of the host computer in Fig. 1.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The invention will now be described in detail hereinbelow on the basis of embodiments shown in the drawings. A printer as an output device and a printer driver (printer manager) as a device driver (program) will be described hereinbelow as an example.

Fig. 1 shows a computer system using the invention. Although the system constructed by a host computer 3000 and a printer (device) 1500 will be shown here as an example, the invention which will be described hereinafter can be also applied to a single apparatus, a system comprising a plurality of apparatuses, and a system in which a printer is connected through a network such as an LAN or the like.

In the host computer 3000 in Fig. 1, reference numeral 1 denotes a CPU. In the embodiment, the CPU 1 processes a document in which figure, image, characters, table (including a table calculation or the like), and the like mixedly exist on the basis of a document processing program or the like stored in an ROM for programs in an ROM 3 or an external memory (hard disk, floppy disk, or the like) 11 and also controls devices connected to a system bus 4.

Reference numeral 2 indicates an RAM which functions as a main memory, a work area, or the like of the CPU 1.

Reference numeral 3 denotes the ROM. A control program (control program shown in flowcharts of Figs. 6 and 7) of the CPU 1, an operating system program, and the like are stored in the program ROM in the ROM 3 or the external memory 11. The ROM 3 has: a portion of an ROM for fonts which stores font data which is used upon document processes and the like; and a portion of an ROM for data for storing various data which is used when the document processes or the like are executed.

Reference numeral 5 denotes a keyboard controller (KBC) for controlling a key input from a keyboard 9 or a pointing device (not shown). Reference numeral 6 denotes a CRT controller (CRTC) for controlling a display of a CRT display (CRT) 10.

Reference numeral 7 indicates a disk controller (DKC) for controlling inputting/outputting operations of data from/to the external memory 11 such as hard disk (HD), floppy disk (FD), or the like to store a boot program, various applications, font data, a user file, an edition file, a printer control command forming program (printer driver) and the like.

Reference numeral 8 denotes a printer controller (PRTC) which is connected to the printer 1500 through a predetermined bidirectional interface (for example, Centronics interface) 21 and executes a controlling process of a communication with the printer 1500.

In the document process of the CPU 1, it is assumed that a graphical interface is used and a control for equalizing an output from the printer 1500 and an output from the CRT 10, namely, what is called a WYSIWYG (What You See Is What You Get) process is executed. In this case, for example, a developing (rasterizing) process of a bit map font or an outline font stored in the font ROM or the like is performed in a display information RAM set on the RAM 2. By transferring bit map data on the display information RAM to the printer 1500 or by using the outline font stored in the printer 1500, the same output can be obtained from the printer 1500 and CRT 10.

In the document process, the CPU 1 opens various windows registered on the basis of commands instructed by a mouse cursor (not shown) or the like on the CRT 10 and executes various data processes.

On the other hand, in the printer 1500, reference numeral 12 denotes a printer CPU for outputting an image signal as output information to a printing section (printer engine) 17 that is connected to a system bus 15 on the basis of a control program stored in an ROM for programs in an ROM 13 and a control program stored in an external memory 14.

Control programs (Figs. 3 and 4) of the CPU 12, which will be described hereinlater, and the like are also stored in the program ROM in the ROM 13 or the external memory 14. The font data which is used when the output information is formed or the like is stored in an ROM for fonts in the ROM 13. In case of a printer having no external memory 14 such as a hard disk or the like, information which is used on the host computer is stored in an ROM for data in the ROM 13. The CPU 12 can execute a communicating process with the host computer through an input section 18 and can notify information in the printer or the like to the host computer 3000.

Reference numeral 19 denotes an RAM which functions as a main memory, a work area, or the like of the CPU 12. A memory capacity of the RAM 19 can be expanded by an optional RAM connected to an extension port (not shown). The RAM 19 is used in an output information developing area, an environm nt data memory area, an NVRAM, or the like.

An access to the external memory 14 such as hard disk (HD), IC card, or the like of the printer 1500 is controlled by a memory controller (MC) 20. The external memory 14 is connected as an option and stores font data, an emulation program, form data, and the like. A switch, an LED display and the like for the user interface are arranged on an operation section 1501.

The number of external memories of the host computer 3000 and the printer is not limited to one but at least one or more external memories can be provided. A plurality of external memories in which an optional font card and a program for interpreting printer control languages of different language systems in addition to the built-in fonts have been stored can be also connected. Further, the printer 1500 can also have an NVRAM (not shown) and printer mode setting information from the operation section 1501 can be also stored.

Further, according to the embodiment, printer drivers corresponding to various output devices such as a printer and the like are provided in the ROM 3 or external memory 11 on the host computer side. The operating system (OS) which controls the host computer is loaded onto the RAM 2 and is executed, thereby controlling the corresponding output device such as a printer or the like.

For example, when a document formed by using an application program or the like on the host computer is printed, by giving various parameters to the printer driver loaded on the RAM 2, a printer control command necessary to print the document is formed and the formed printer control command is transferred to the printer through the interface 21, thereby printing.

In the embodiment, such a printer driver is prepared for the printer 1500. That is, the printer driver corresponding to the specification of the self machine is stored in the ROM 13 or external memory 14 of the printer 1500 and is up-loaded to the host computer 3000 and is used. Preferably, various printer driver programs for various OSs which correspond to the specification of the self machine are stored and a proper printer driver program is enabled to be up-loaded in accordance with the kind of OS that is notified from the host computer 3000 as will be described hereinlater.

Hitherto, for example, in case of printing the document formed by using the application program or the like on the host computer, when the printer driver for forming the control command of the connected printer is not preliminarily prepared in the ROM 3 or external memory 11 (hard disk), the printer driver is supplied to the user from the external memory 11 such as a floppy disk or the like. After that, the user designates the printer driver to the OS by the set file or a predetermined operation, thereby completing the installation of the printer driver for the first time.

According to the embodiment, however, the OS first obtains data for identifying the printer from the printer through the bidirectional interface 21. When the printer driver for the printer is not stored in the ROM 3 or external memory 11, the printer driver stored in the ROM 13 of the printer is further obtained and is stored into the external memory 11. After that, the printer driver is loaded to the RAM 2 and is executed or the printer driver is directly loaded to the RAM 2 and is executed, thereby forming the printer control commands necessary for printing the documents. The printer control commands can be transferred to the printer through the interface 21.

An embodiment of the invention will be described in detail hereinbelow with reference to the flowcharts and the like.

Fig. 2 shows a construction of the printer 1500 in more detail. In Fig. 2, the printer drivers which are different according to the OSs have been stored in the ROM 13 or external memory 14. The printer drivers corresponding to various OSs which are major in the market are prepared as many as possible and are stored in the ROM 13 or external memory 14. In Fig. 2, the printer drivers for a plurality of OSs are stored in both of the ROM 13 and external memory 14. The data formats of the printer drivers can be a loadable module format or an executable format as long as they are formed in accordance with the specifications of the corresponding OSs.

In the invention, the OS which controls the host computer to which the printer is connected is specified through the bidirectional interface 21 and the printer driver for the OS can be transferred by an instruction of the OS. When transferring the printer driver, it is also possible that the printer driver is once read into the RAM 19 on the printer and is transferred. The printer driver can be directly transferred from the ROM 13 or external memory 14 to the input section 18 if it is possible.

Fig. 3 shows a processing procedure for the reception data on the printer 1500 side. The processes are executed by the CPU 12 of the printer. In step S31 in Fig. 3, data transmitted through the bidirectional interface 21 is read out from the input section 18 and is transferred to the RAM 19.

In step S32, a check is made to see if the data is an identification data request command for identifying the printer of the OS. If YES, the printer identification data is transmitted to the host computer (step S33). Since the identification data request command for identifying the printer is transmitted to the printer 1500 at the stage when a formal printer driver corresponding to the printer 1500 does not (may not) operate, a relatively simple format (for instance, a predetermined escape sequence or the like) such that the command can be transmitted by directly controlling a port to which the printer is connected without passing through the printer driver is preferable. The printer identification data has preliminarily been stored in the data ROM 13 or external memory 14.

A check is made to see if the received data is a transmission command of the printer driver in step S34. If YES, the printer driver as a program for forming the printer control command is transmitted as data to the host computer (step S35).

When the received data is not a port identification data request nor the printer driver transmission command, it is judged that the received data is an ordinary print control command and a printing process is exe-

cuted (step S36). The printer repeatedly executes the above processes.

Fig. 4 is a flowchart showing a processing procedure for transferring the printer driver in step S35 in Fig. 3. The processes are executed by the CPU 12 of the printer. It is now assumed that the printer driver transmission command judged in step S34 includes identification data which can identify the OS. The identification data of the OS is taken out in step S41.

A check is made in step S42 to see if the printer driver for the OS designated by the identification data exits in the ROM 13 or external memory 14. If the printer driver doesn't exist (step S43), data indicative of an error is transmitted to the host computer. When the designated printer driver exists, the printer driver corresponding to the OS identified in step S41 is transferred from the ROM 13 or external memory 14 to the RAM 19 (Steps S44, S45) and is transferred to the host computer 3000 through the bidirectional interface 21.

Fig. 5 shows the main part of the host computer 3000 in detail. As shown in the diagram, the printer drivers which are different according to the printers or printer control languages have already been stored in the ROM 3 or external memory 11.

In the invention, the printer connected to the host computer is specified through the bidirectional interface 21 and the printer driver needed by the OS can be transferred to the printer. As mentioned above, when the printer driver is uploaded from the printer, the received printer driver can be once read into the RAM 2 on the host computer and, after that, stored into the external memory 11 or it can be also directly transferred to the external memory 11 if it is possible. The printer driver is loaded into the RAM 2 by an arbitrary proper method upon execution. When the printer driver is transferred from the printer to the host computer, the printer driver can be also compressed in a predetermined format and transferred. By such operations, a transfer speed between the printer and the host computer is improved.

Fig. 6 shows a using method of the printer driver in the OS of the host computer. Processes of Fig. 6 are executed by the CPU 1 of the host computer.

In step S61 in Fig. 6, an identification data request command for identifying the printer is transmitted to the printer that is connected directly (or through a network or the like) at present (refer to steps S31 to S33 in Fig. 3). In step S62, the printer identification data returned from the printer is analyzed. The processes (steps S61 and S62) need to be executed at the time of the activation of the host computer or the printer, or when a connecting state of the device such as a printer 1500 or the like is updated, or the like.

In step S63, a check is made to see whether the printer driver for the identified printer exists in the ROM 3 or external memory 11. When the printer driver for the identified printer exists, the printer driver is transferred from the ROM 3 or external memory 11 to th RAM 2 and is outputted to the printer by using the printer driver. In this instance, the printer driver executes a process such that a drawing command used in the OS or application is converted to the printer control command (page description language) that can be understood by the printer, or the like.

When no printer driver exists in step S63, a printer driver transmission command is transmitted to the printer (step S64), the printer receives the transmitted printer driver in step S65, and the printer driver is stored into the external memory 11 (step S66). In step S67, the printer driver is loaded into the RAM 2, the drawing command formed on the basis of the application program is converted to the printer control command by using the printer driver, and the printer control command is outputted to the printer.

With respect to the transfer of the printer driver from the external memory 11 or the like to the RAM 2 in step S67, it is also possible to preliminarily directly transfer the printer driver from the PRTC 8 when it is received in step S66. By using the printer driver transferred onto the RAM 2, print data is converted to a printer print control command, thereby performing the printing.

According to the above embodiment, when the printer driver for the connected printer doesn't exist on the host computer, the printer driver built in the printer is transferred through the bidirectional interface, thereby enabling the printer driver to be automatically supplied to the OS without troubling the user.

That is, since the printer drivers for the self machine as many as the number of various OSs are prepared for the printer and a desired printer driver is transferred to the host as necessary and can be used by the OS of the host, the correct printer driver can be automatically loaded to the OS without designating the printer driver by the user.

In case of selecting the printer driver, since an intervention of the user is unnecessary, the printer driver is not erroneously used, so that the printing of a high quality can be always performed.

According to the embodiment, when the printer driver is up-loaded from the printer, by storing the printer driver into the external memory (such as a hard disk drive or the like), the up-loading of the printer driver is unnecessary from the next time on. As shown in Fig. 7, however, there is also considered a process of up-loading the printer driver every printing, namely, dynamically replacing the printer driver.

Fig. 7 shows processes on the host side. In step S71 in Fig. 7, a data request command for identifying the printer is transmitted to the printer connected directly or through the network at present. In step S72, the printer identification data returned by the printer is analyzed. Although the processes (steps S71 and S72) are executed when the host computer is activated, when the printer is activated, when their connecting states are updated, or the like, they have to be also executed when the printing is executed and when an inquiry of information depending on the printer such as a font for the WYSIWYG process is necessary.

In step S73, a check is made to see if the printer driver for the identified printer exists in the ROM 3 or external memory 11. If YES, the printer driver is transferred from the ROM 3 or external memory 11 to the RAM 2, the drawing command is converted to the printer control command by using the transferred printer driver and the printer control command is outputted to the printer. When the printer driver doesn't exist, the printer driver transmission command is transmitted to the printer (step S74), the data returned from the printer is received in step S75 and the printer driver is developed in the RAM 2 (step S76).

That is, different from Fig. 6, the printer driver is not stored into the external memory 11 but is directly developed in the RAM 2, the program is executed, the drawing command is converted to the printer control command, and the printer control command is outputted to the printer in Fig. 7. Consequently, the printer driver can be dynamically replaced without storing and holding the printer driver onto the host computer and a consumption of the external memory can be suppressed. For example, the consumption of the external memory by the printer driver of the printer which is rarely used can be suppressed, so that the external memory can be provided for storing the user data.

Although the printer has been described above as an embodiment, a similar construction can be also realized even in case of device drivers such as a display, a plotter, a sound card, and other peripheral devices. The interface between the peripheral device and the host is not limited to the printer port but any device such as serial port, SCSI port, extension bus, or the like can be also used. It is sufficient to construct in a manner such that the device driver of the peripheral device is prepared for the peripheral device and the device driver is transferred to the host as necessary and is loaded to the OS of the host.

As will be obviously understood from the above description, according to the invention, there is provided the information processing apparatus which is constructed by the host computer and the peripheral device connected to the host computer through the bidirectional interface and which inputs and outputs data between the host computer and the peripheral device through the device driver that corresponds to the peripheral device and is loaded in the OS of the host computer, wherein the peripheral device has memory means in which the device driver corresponding to the peripheral device has been stored, the device driver in the memory means is transferred to the host computer as necessary and is loaded into the OS of the host computer. Thus, the device driver prepared for the peripheral device can be transferred to the host computer as necessary and automatically loaded into the OS, the OS can automatically use the correct device driver adaptive to the connected peripheral device without troubling the user, and it can be prevented that the device drive is erroneously used. By dynamically transferring the device driver from the peripheral device to the host computer, the consumption of the external memory device of the host computer can be suppressed.

There is provided an information processing apparatus which is constructed by a host computer and a peripheral device connected to the host computer through a bidirectional interface and which inputs and outputs data between the host computer and the peripheral device through a device driver that corresponds to the peripheral device and is loaded in the OS of the host computer. An output device such as a printer which is controlled by the host computer has an identifying unit to identify the kind of OS of the host computer and a controller to output a device driver for controlling the device to the host computer on the basis of the identification result. The peripheral device has a memory in which the device driver corresponding to the peripheral device has been stored, the device driver in the memory is transferred to the host computer as necessary and is loaded into the OS of the host computer.

Claims

1. A device which is controlled by an information processing apparatus, characterized by:
   means for identifying a kind of an OS of said information processing apparatus; and
   control means for outputting a device driver for controlling said device to said information processing apparatus on the basis of an identification result of said identifying means.

2. A device according to claim 1, characterized in that said identifying means identifies the kind of the OS of said information processing apparatus on the basis of OS identification information included in a printer driver output command.

3. A device according to claim 1, characterized in that said control means selects the printer driver corresponding to the OS identified by said identifying means.

4. A device according to claim 3, characterized in that when the printer driver corresponding to the OS identified by said identifying means cannot be selected, said control means notifies such a fact to said information processing apparatus.

5. A device according to claim 1, characterized in that said device is a printer.

6. A device according to claim 1, characterized in that said information processing apparatus is a host computer.

7. A device according to claim 1, characterized in that said information processing apparatus and said device are connected through a bidirectional interface.

FIG. 1

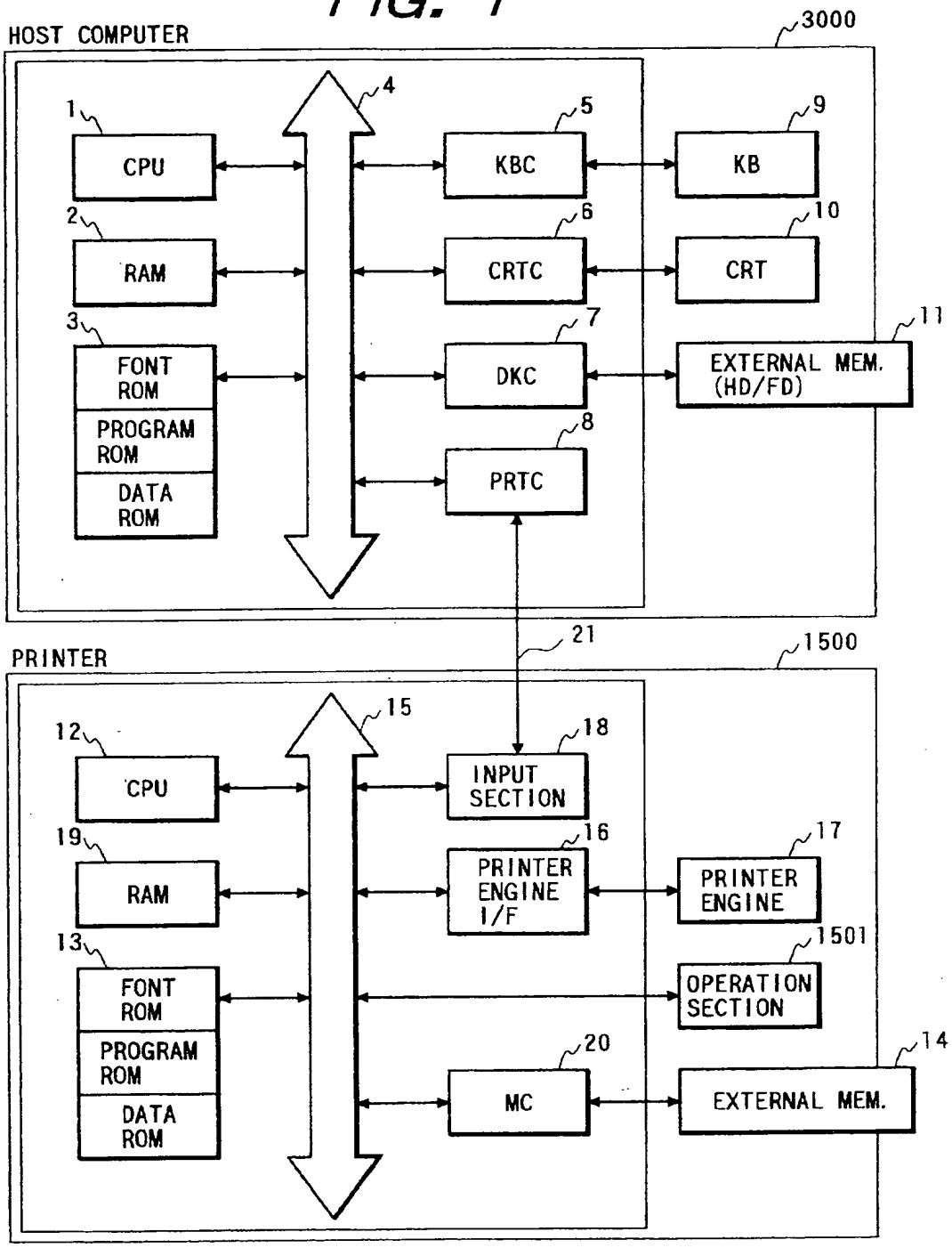

HOST COMPUTER
3000
1
CPU
2
RAM
3
FONT ROM
PROGRAM ROM
DATA ROM
4
5
KBC
6
CRTC
7
DKC
8
PRTC
9
KB
10
CRT
11
EXTERNAL MEM. (HD/FD)
21
PRINTER
1500
12
CPU
19
RAM
13
FONT ROM
PROGRAM ROM
DATA ROM
15
18
INPUT SECTION
16
PRINTER ENGINE I/F
17
PRINTER ENGINE
1501
OPERATION SECTION
20
MC
14
EXTERNAL MEM.

FIG. 2

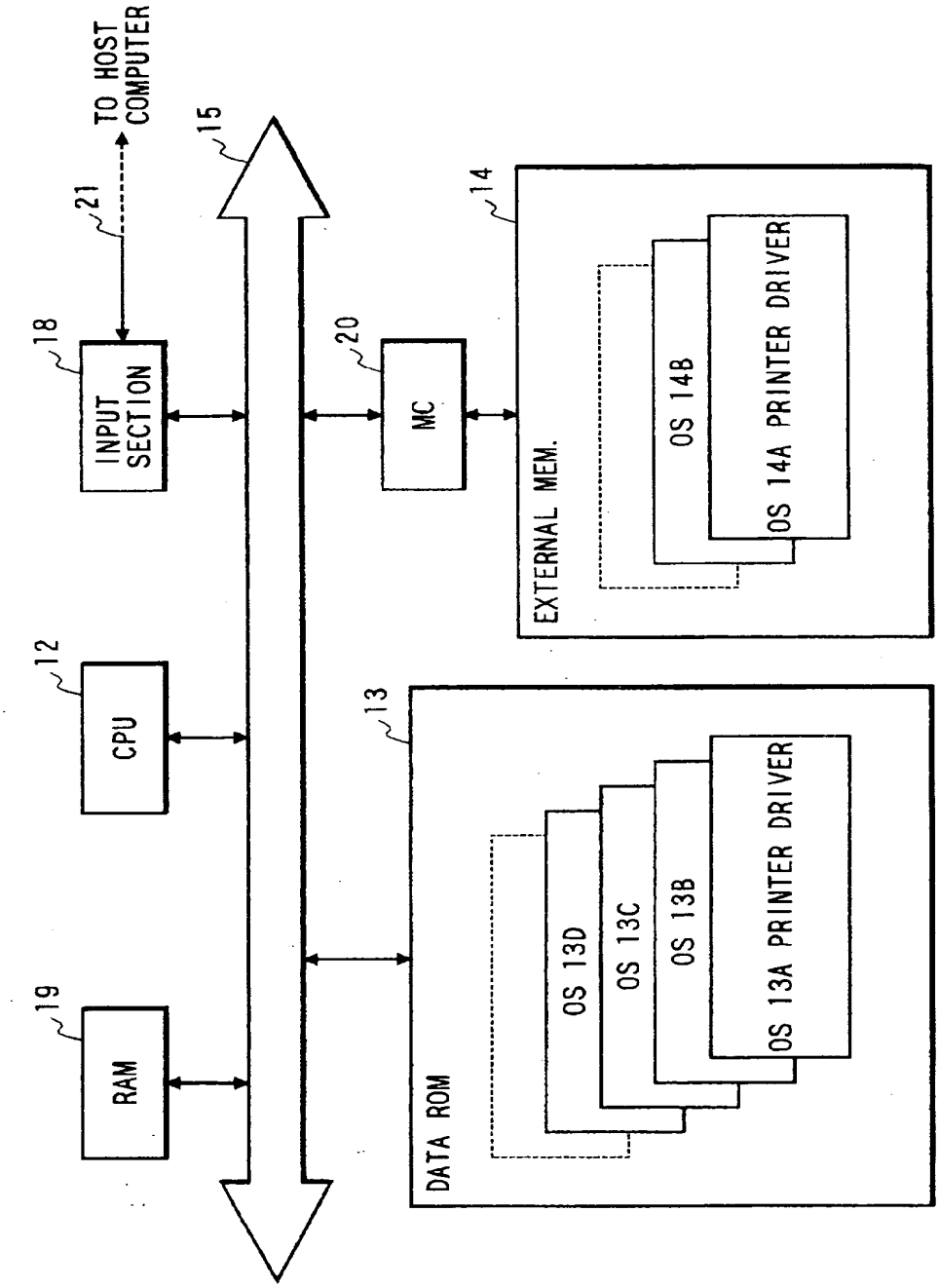

TO HOST COMPUTER
21
15
14
18
20
INPUT SECTION
MC
EXTERNAL MEM.
OS 14B
OS 14A PRINTER DRIVER
12
CPU
13
DATA ROM
OS 13D
OS 13C
OS 13B
OS 13A PRINTER DRIVER
19
RAM

FIG. 3

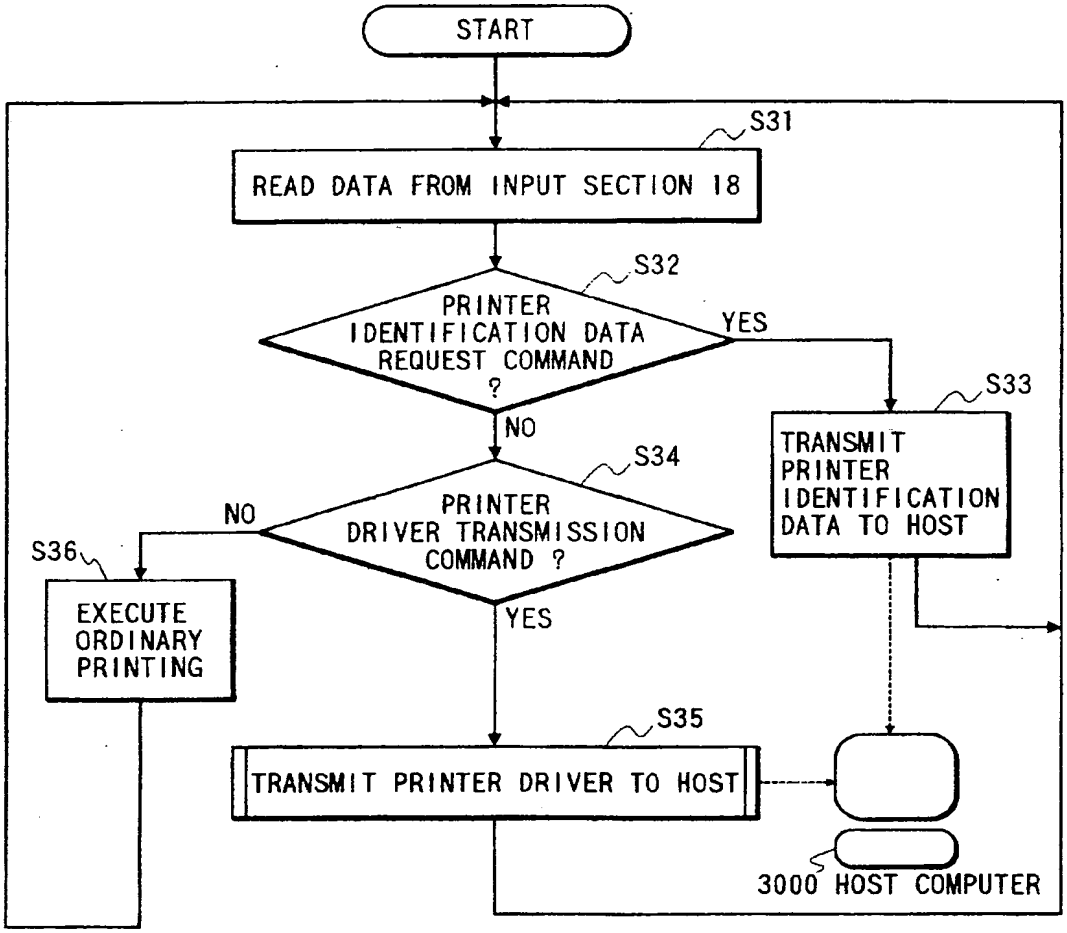

START
S31
READ DATA FROM INPUT SECTION 18
S32
PRINTER IDENTIFICATION DATA REQUEST COMMAND ?
YES
NO
S33
TRANSMIT PRINTER IDENTIFICATION DATA TO HOST
S34
PRINTER DRIVER TRANSMISSION COMMAND ?
NO
S36
EXECUTE ORDINARY PRINTING
YES
S35
TRANSMIT PRINTER DRIVER TO HOST
3000 HOST COMPUTER

FIG. 4

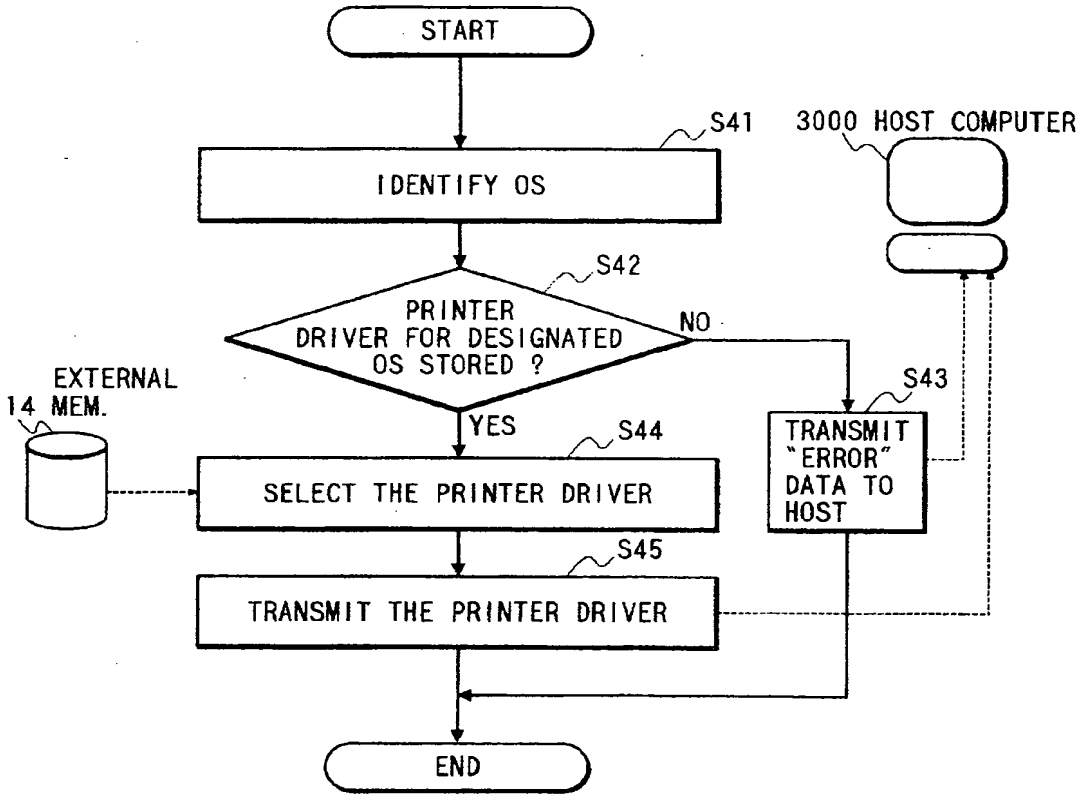

START
S41
IDENTIFY OS
3000 HOST COMPUTER
S42
PRINTER DRIVER FOR DESIGNATED OS STORED ?
NO
YES
S43
TRANSMIT "ERROR" DATA TO HOST
EXTERNAL 14 MEM.
S44
SELECT THE PRINTER DRIVER
S45
TRANSMIT THE PRINTER DRIVER
END

FIG. 5

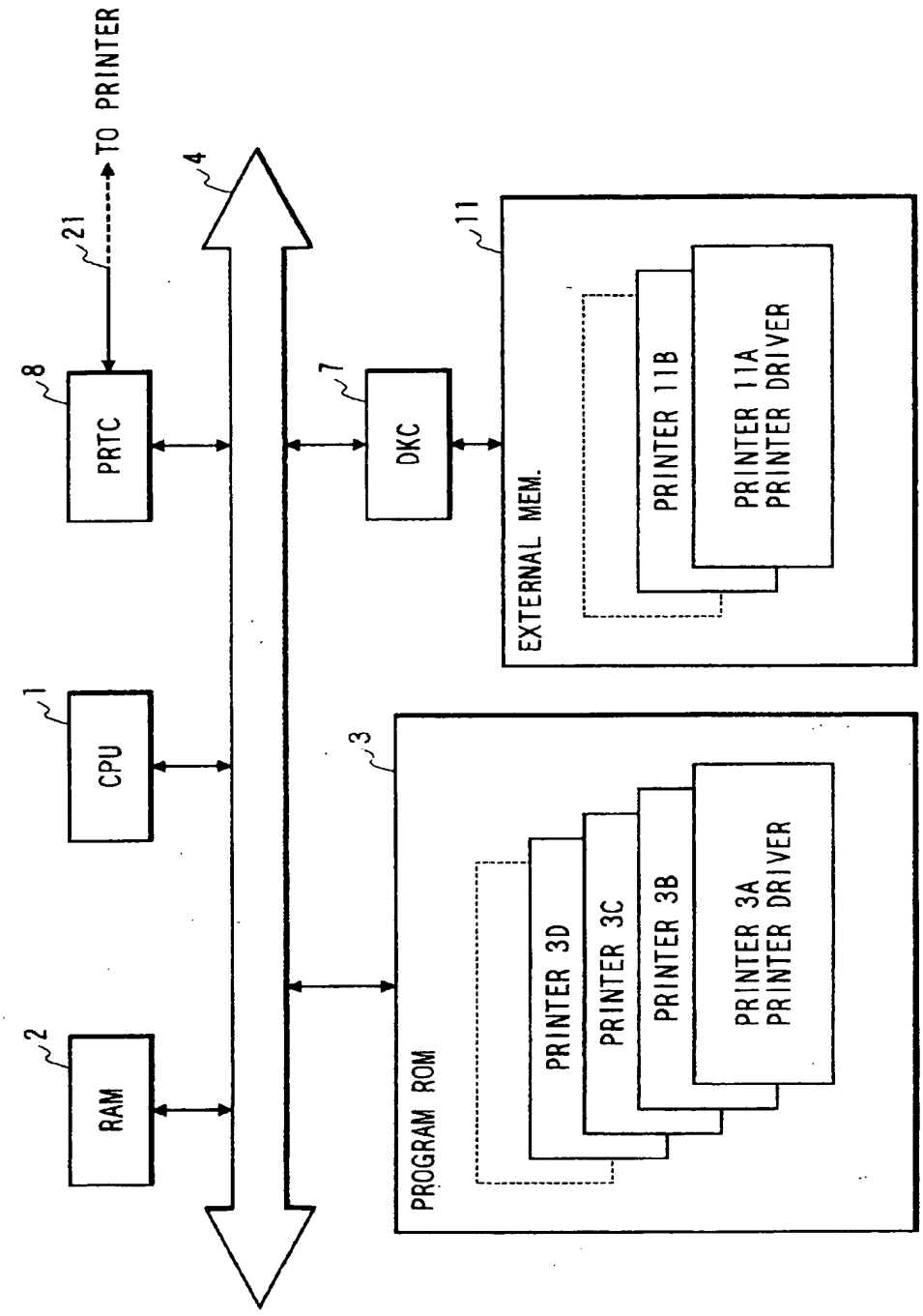

TO PRINTER
21
4
11
8
7
PRTC
DKC
EXTERNAL MEM.
PRINTER 11B
PRINTER 11A
PRINTER DRIVER
1
3
CPU
PROGRAM ROM
PRINTER 3D
PRINTER 3C
PRINTER 3B
PRINTER 3A
PRINTER DRIVER
2
RAM

FIG. 6

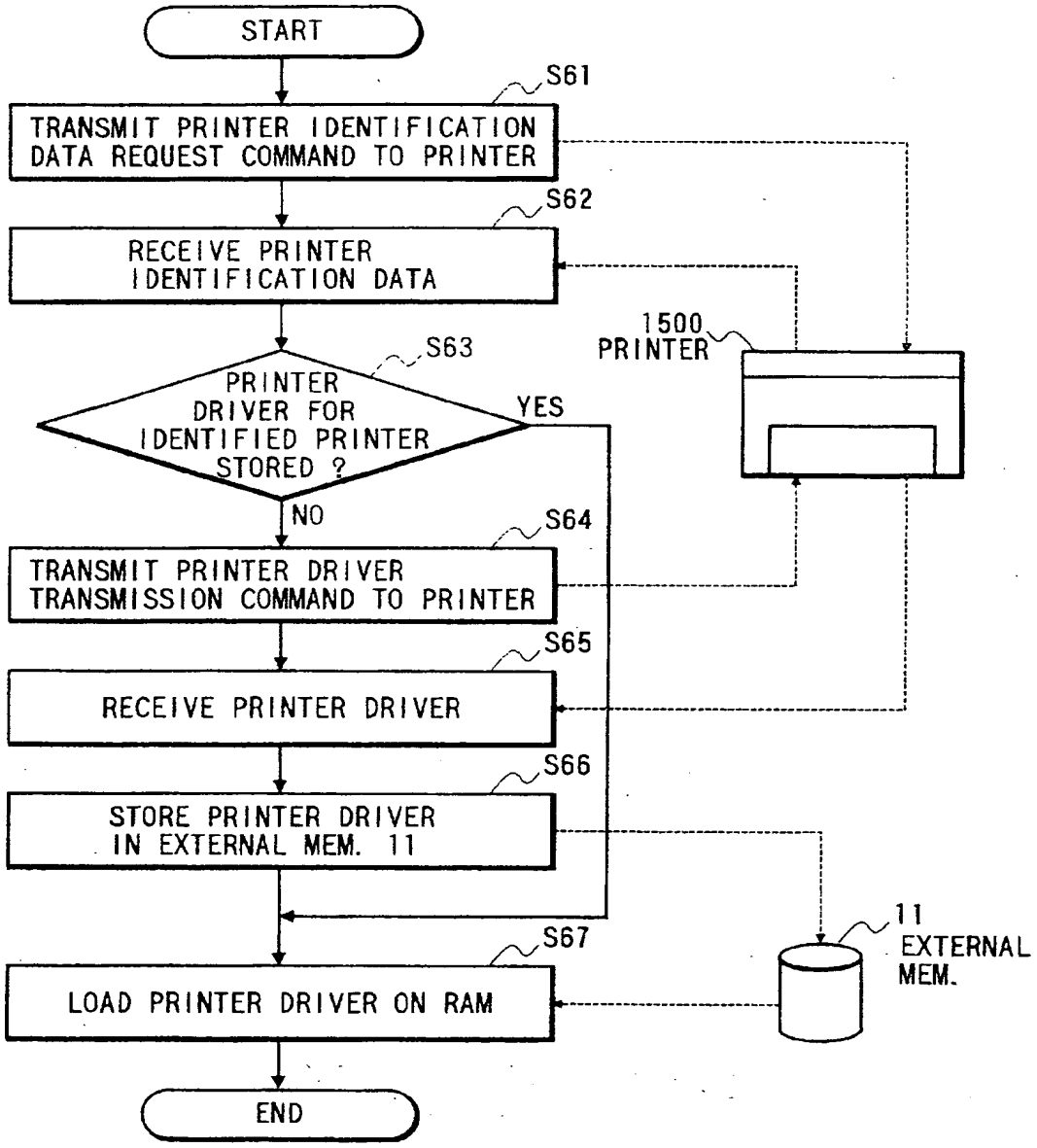
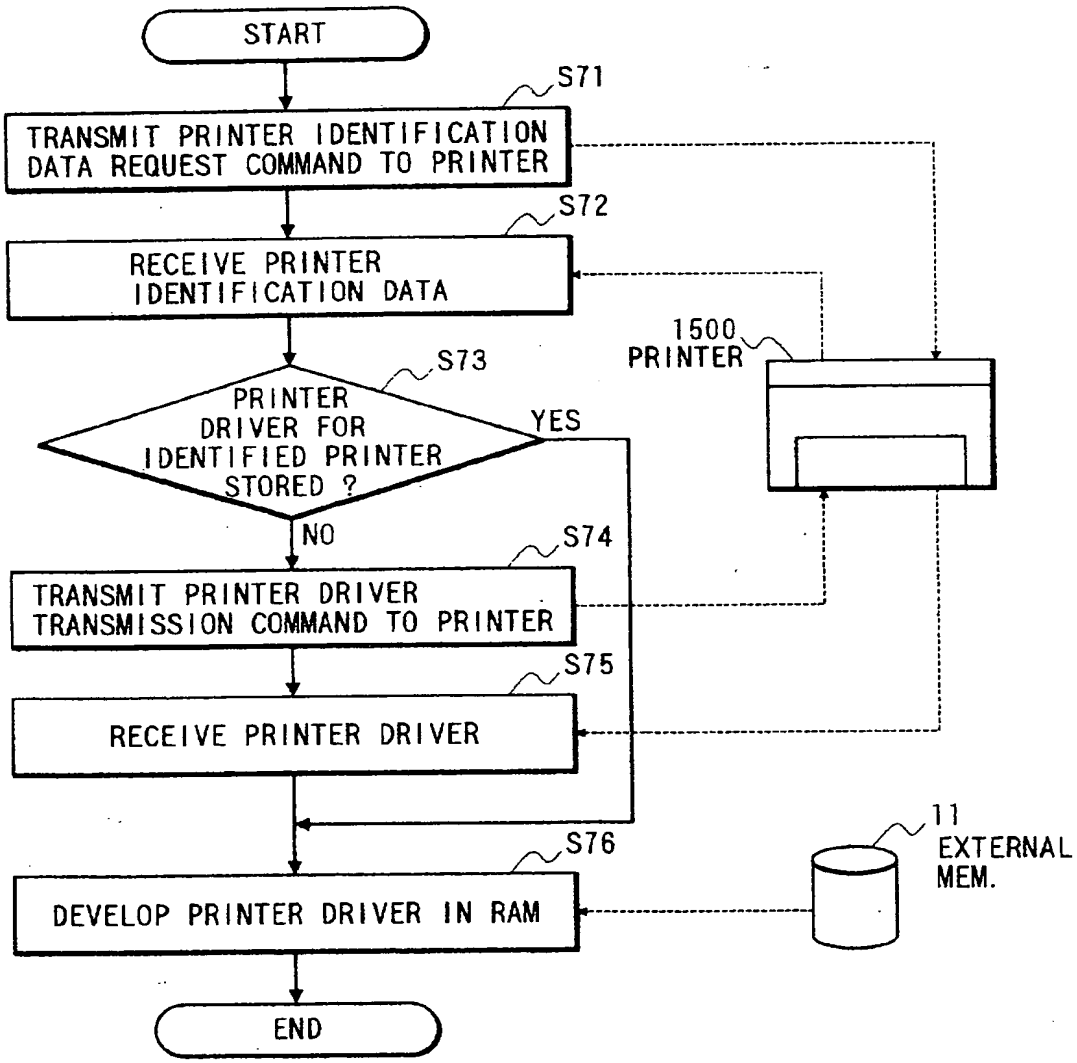

START
S61
TRANSMIT PRINTER IDENTIFICATION DATA REQUEST COMMAND TO PRINTER
S62
RECEIVE PRINTER IDENTIFICATION DATA
1500
PRINTER
S63
PRINTER DRIVER FOR IDENTIFIED PRINTER STORED ?
YES
NO
S64
TRANSMIT PRINTER DRIVER TRANSMISSION COMMAND TO PRINTER
S65
RECEIVE PRINTER DRIVER
S66
STORE PRINTER DRIVER IN EXTERNAL MEM. 11
11
EXTERNAL MEM.
S67
LOAD PRINTER DRIVER ON RAM
END

FIG. 7


START
S71
TRANSMIT PRINTER IDENTIFICATION DATA REQUEST COMMAND TO PRINTER
S72
RECEIVE PRINTER IDENTIFICATION DATA
1500
PRINTER
S73
PRINTER DRIVER FOR IDENTIFIED PRINTER STORED ?
YES
NO
S74
TRANSMIT PRINTER DRIVER TRANSMISSION COMMAND TO PRINTER
S75
RECEIVE PRINTER DRIVER
11
EXTERNAL MEM.
S76
DEVELOP PRINTER DRIVER IN RAM
END